CLAIMS

WE CLAIM:

1. A method of processing multimedia data, the method comprising:

creating a topology of connections between one or more multimedia components in a topology generating element, the topology describing a set of input multimedia streams, one or more sources for the input multimedia streams, a sequence of operations to perform on the multimedia data, and a set of output multimedia streams;

transmitting the topology to a media processor; and

passing data according to the topology, the passing governed by the media processor.

2. The method of claim 1 further comprising performing the sequence of multimedia operations on the multimedia data to create the set of output multimedia streams.

3. The method of claim 1 wherein the multimedia components are software objects.

4. The method of claim 1 wherein the topology generating element is a topology loader.

5. The method of claim 1 wherein the topology generating element is an application program.

6. The method of claim 1 wherein the media processor exposes the multimedia data to an application.

7. The method of claim 1 wherein the media processor accepts the multimedia data via being configured as a media sink.

8. A system for processing multimedia data, the system comprising:

a control layer configured to receive instructions from an application, the control layer including:

a topology generating element configured to generate a topology describing a set of input multimedia streams, one or more sources for the input multimedia streams, a

sequence of operations to perform on the multimedia data, and a set of output multimedia streams; and

a media processor configured to govern the passing of the multimedia data as described in the topology and govern the performance of the sequence of multimedia operations on the multimedia data to create the set of output multimedia streams;

a core layer coupled to the control layer, the core layer configured to include:

the input media streams;

the sources for the input multimedia streams;

one or more transforms configured to operate on the multimedia data;

one or more stream sinks coupled to the control layer; and

one or more media sinks configured to provide the set of output multimedia streams.

9.   The system of claim 8 wherein the multimedia components are software objects.

10.   The system of claim 8 wherein the topology generating element is a topology loader.

11.   The system of claim 8 wherein the topology generating element is an application program.

12.   The system of claim 8 wherein the media processor exposes the multimedia data to an application.

13.   The system of claim 8 wherein the media processor accepts the multimedia data via being configured as a media sink.

14.   A method of changing a first topology in use by a media processor while the media processor is active, the method comprising:

preserving the present state of the media processor;

receiving one or more instructions to convert the first topology into a second topology; and

updating the first topology to the second topology in accordance with the one or more instructions.

15. The method of claim 14 wherein the one or more instructions contain the difference between the first topology and the second topology.

16. The method of claim 14 wherein the media processor resumes the interface activity after updating the first topology to the second topology.

17. The method of claim 16 wherein the media processor sends messages to an application upon resuming interface activity.

18. The method of claim 16 wherein the media processor allows message calls until a topology change is complete.

19. The method of claim 14 wherein the media processor receives a message from an external source to initiate the process of changing the first topology.

20. A method of determining how to use a set of multimedia components to perform a sequence of multimedia operations on one or more streams of multimedia data in a media processor, the method comprising:
locating one or more multimedia components with outputs connected to an input of a sink device;
querying the multimedia components to determine if a sample is available, the querying including checking inputs to the multimedia components if a sample is not available;
if the inputs do not have a sample available, checking a media source feeding the multimedia components for a sample;
if the media source does not have a sample available, performing an end of file function or declaring an error condition;
if a sample is available, moving the sample to a next multimedia component of the multimedia components.

21. A method for retrieving a section of a media stream, the method comprising:
caching the section of a media stream, the cached section of the media stream containing a presentation point of the media stream;

receiving a request from an external source to the media processor to retrieve the cached section of the media stream;

searching to identify whether the section of the media stream was cached; and

if the section of the media stream was cached, transferring the requested cached section.

22. The method of claim 21 wherein the section of media stream is cached according to user settings in an application.

23. The method of claim 21 wherein the external source is an application program.

24. The method of claim 21 wherein a number of samples contained in the cached section is programmable.

25. The method of claim 21 wherein the media stream represents video data.

26. The method of claim 21 wherein a number of frames contained in the cached section is programmable.

27. The method of claim 21 wherein the cached section of media data is continuous.

28. A computer readable medium having stored therein instructions for performing acts for processing multimedia data, the acts comprising:

creating a topology of connections between one or more multimedia components in a topology generating element, the topology describing a set of input multimedia streams, one or more sources for the input multimedia streams, a sequence of operations to perform on the multimedia data, and a set of output multimedia streams;

transmitting the topology to a media processor; and

passing data according to the topology, the passing governed by the media processor.

29. The computer readable medium of claim 28 further comprising performing the sequence of multimedia operations on the multimedia data to create the set of output multimedia streams.

30. The computer readable medium of claim 28 wherein the multimedia components are software objects.

31.   The computer readable medium of claim 28 wherein the topology generating element is a topology loader.

32.   The computer readable medium of claim 28 wherein the topology generating element is an application program.

33.   The computer readable medium of claim 28 wherein the media processor exposes the multimedia data to an application.

34.   The computer readable medium of claim 28 wherein the media processor accepts the multimedia data via being configured as a media sink.

35.   A computer readable medium having stored therein instructions for performing acts for changing a first topology in use by a media processor while the media processor is active, the acts comprising:
halting interface activity in the media processor;
preserving the present state of the media processor;
receiving one or more instructions to convert the first topology to a second topology; and
updating the first topology to the second topology in accordance with the one or more instructions, the media processor continuing processing the first topology until each multimedia component called by the first topology is in a state to allow the first topology to be changed.

36.   The computer readable medium of claim 35 wherein the one or more instructions contain the difference between the first topology and the second topology.

37.   The computer readable medium of claim 35 wherein the multimedia components include at least a media source and a media transform.

38.   The computer readable medium of claim 35 wherein the media processor further resumes the interface activity after updating the first topology to the second topology.

39.   The computer readable medium of claim 38 wherein the media processor further sends messages to an application upon resuming interface activity.

40. The computer readable medium of claim 35 wherein the media processor receives a message from an external source to initiate the process of changing the first topology.

41. A computer readable medium having stored therein instructions for performing acts for determining how to use one or more multimedia components to perform operations on multimedia data in a media processor, the acts comprising:
locating the one or more multimedia components that are directly generating multimedia samples for a sink device;
querying the multimedia components to determine if a sample is available, the querying including checking inputs to the objects if a sample is not available;
if the inputs do not have a sample available, checking a source feeding the objects for a sample;
if the source does not have a sample available, performing an end of file functionality or declaring an error condition;
if a sample is available, moving the sample to a second object.

42. The computer readable medium of claim 41 wherein the multimedia components are specified in a topology.

43. A computer readable medium having stored therein instructions for performing acts for retrieving a section of a media stream, the acts comprising:
caching the section of a media stream, the cached section of the media stream containing a presentation point of the media stream;
receiving a request from an external source to the media processor to retrieve the cached section of the media stream;
searching to identify whether the section of the media stream was cached;
if the section of the media stream was cached, transferring the requested cached section.

44. The computer readable medium of claim 43 wherein the section of media stream is cached according to user settings in an application.

45. The computer readable medium of claim 43 wherein the

46. The computer readable medium of claim 43 wherein the external source is an application program.

47. The computer readable medium of claim 43 wherein a number of samples contained in the cached section is programmable.

48. The computer readable medium of claim 43 wherein the media stream represents video data.

49. The computer readable medium of claim 43 wherein a number of frames contained in the cached section is programmable.

50. The computer readable medium of claim 43 wherein the cached section of media data is continuous.